

**Информационные технологии от первого лица**

**А. Н. Терехов**

# **ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ**

**Учебное пособие**

**Рекомендовано учебно-методическим объединением по образованию в области инновационных междисциплинарных образовательных программ в качестве учебного пособия по специальности «Математическое обеспечение и администрирование информационных систем» – 010503**



**Интернет-Университет  
Информационных Технологий  
[www.intuit.ru](http://www.intuit.ru)**



**БИНОМ.  
Лаборатория знаний [www.lbz.ru](http://www.lbz.ru)**

**Москва  
2006**

ОБОРОТ ТИТУЛА

## Предисловие

Основу данной книги составляет курс лекций, который вот уже много лет я читаю на отделении информатики математико-механического факультета СПбГУ. Технология промышленного программирования — не совсем обычная тема для мат-меха, сама постановка такого курса была вызвана несколькими причинами:

- было замечено, что многие выпускники не могут сразу приступить к производительной работе, не понимают необходимости жесткого планирования и не менее жесткой отчетности, не чувствуют особенностей работы в команде, не знают, что такое бюджет и т.д.
- преподаватели университета часто не имеют опыта работы в промышленности, поэтому учат студентов на академическом уровне, «по книжкам»;
- наконец, первые же столкновения с промышленными задачами показали, что промышленные технологии требуют совершенно иных приемов программирования, других инструментальных средств и организационных методик. Именно по этим причинам возникла новая наука — software engineering (программная инженерия), сильно отличающаяся от ставшей уже традиционной computer science.

Так получилось, что с 1980 года я стал работать с военными организациями Ленинграда. Первоначально это была настоящая «принудилровка», похожая на столь же обязательную в те времена «картошку». Только через несколько лет я понял, что судьба преподнесла мне подарок, дав возможность поработать с новыми людьми, задачами, с совершенно необычными для университетов требованиями и технологиями. В то время я возглавлял лабораторию системного программирования НИИ ММ ЛГУ, в которой работало около 15 человек. Нашему коллективу пришлось принять вызов и заняться исследованиями и разработками в новой для себя области. Мы успешно решили множество важных государственных задач, разработали и внедрили промышленные технологии, воспитали большое количество специалистов. Надеюсь, после этих слов ни у кого не возникнет вопрос, почему это университетский профессор читает лекции и пишет книгу о промышленном программировании.

## Благодарности

Прежде всего я хочу поблагодарить уже ушедших из жизни капитана I ранга, доктора технических наук, профессора Владимира Петровича Морозова и академика Андрея Петровича Ершова. В.П. Морозов был первым, кто пришел в наш коллектив с задачами промышленного программирования, очень во многих случаях он опередил свое время. Такие его идеи как контроль за ходом разработки, графическая схема ситуаций, динамическая группа и др. в начале 1980-х годов было трудно оценить.

Я счастлив, что в отличие от многих коллег, сумел разглядеть за «командирским голосом» зерна будущих технологий.

А.П. Ершов вообще был моим «крестным отцом». Чем-то я ему понравился, во всяком случае, много лет он предлагал мне интересные задачи, вел со мной увлекательные личные беседы, вовлекал во многие свои начинания, например, школьную информатику. Он был председателем Рабочей группы по Алголу 68, а я был его замом. 4-5 раз в год мы собирались в разных городах СССР, общались на самые разные темы, часто далекие от Алгола 68. Каждое собрание кончалось традиционной индейкой – А.П. Ершов был самым активным участником и этого события. Когда передо мной встала зловещая, как мне тогда казалось, перспектива работы с военными, первым, к кому я обратился за советом, был А.П. Ершов. Он поддержал меня, назвал нескольких известных ученых, которые были генералами и адмиралами, чего я, конечно, не знал. Более того, позднее он принял участие в наших работах.

Я благодарен всем участникам нашего, сильно разросшегося, коллектива. Промышленная технология – дело коллективное, в одиночку тут ничего не сделать.

Благодарю В.И. Кияева за материалы лекций по качеству ПО, моего коллегу из США Лена Эрлиха, который как-то раз прочел моим студентам замечательную лекцию по CASE-технологиям. Всегда интересен взгляд на известные вещи с другой стороны.

Но больше всего я благодарен своим детям – Андрею Терехову (мл.) и Карине Тереховой, которые когда-то были моими учениками, а теперь уже много лет являются моими главными критиками, рецензентами и советчиками.

## Об авторе

### **Терехов Андрей Николаевич**

Родился 3 сентября 1949 года. В 1971 г. с отличием окончил Ленинградский государственный университет по специальности «Математическое обеспечение ЭВМ».

Начиная с 1971 года прошел путь от младшего научного сотрудника до заведующего лабораторией системного программирования ВЦ ЛГУ. В 1984 году организовал совместный комплексный отдел Минобразования РСФСР и Минпромсвязи СССР, на базе которого был создан НИИ «Звезда» ЛНПО «Красная заря». В этом НИИ А.Н. Терехов был назначен заместителем директора (по совместительству). В 1991 году основал и возглавил ГП «Терком». С 1996 году возглавил созданную им кафедру системного программирования СПбГУ. С момента создания в 1998 году ЗАО «Ланит-Терком» — генеральный директор компании. В 2002 году организовал и возглавил НИИ информационных технологий СПбГУ. В сентябре 2004 года избран председателем правления ассоциации разработчиков программного обеспечения РУССОФТ, созданной на базе консорциума ФОРТ-РОСС и ассоциации РУССОФТ.

Доктор физико-математических наук, профессор, академик Международной академии информатизации. Член ACM и IEEE Computer Society.

Женат, отец двоих детей. Дочь Карина окончила с отличием математико-механический факультет СПбГУ и аспирантуру Оксфордского университета, там же защитила докторскую диссертацию. В настоящее время — менеджер высокого уровня в компании Nokia. Сын закончил с отличием математико-механический факультет СПбГУ, кандидат физ.-мат. наук, менеджер по связям с университетами России в компании Microsoft (Москва).

Хобби — баскетбол, водный туризм.

## Содержание

<b>Часть I. Общие вопросы технологии программирования.</b> . . . . .	9
1. Понятие технологии программирования . . . . .	9
2. Жизненный цикл программы . . . . .	14
3. Постановка задачи. Оценка осуществимости . . . . .	17
4. Планирование . . . . .	20
5. Управление. . . . .	26
6. Тестирование, обеспечение качества . . . . .	30
7. Групповая разработка, управление версиями. . . . .	35
8. Психология программирования . . . . .	38
9. Организация коллектива разработчиков . . . . .	41
10. Организация коллектива разработчиков в компании Microsoft. . . . .	44
11. Документирование . . . . .	49
12. Сопровождение . . . . .	51
13. Реинжиниринг. . . . .	53
14. Управление качеством. . . . .	56
15. Стандарты ISO . . . . .	63
16. Capability Maturity Model for Software (Модель SEI SW-CMM) . . . . .	67
17. CASE-технологии . . . . .	73
<b>Часть II. Технология программирования встроенных систем реального времени.</b> . . . . .	80
1. Понятие встроенной системы реального времени. . . . .	80
2. Инструментальная и целевая ЭВМ . . . . .	82
3. Комплекс вычислительных средств . . . . .	84
4. Параллельные процессы. . . . .	89
5. Работа с временными интервалами. . . . .	93
6. Организация вычислительного процесса . . . . .	95
7. Технология RTST. . . . .	99
8. Технология REAL . . . . .	106
Литература . . . . .	110

<b>Приложения: интервью, статьи, доклады, об авторе</b> . . . . .	113
Как готовить системных программистов . . . . .	114
Индустриальная программа подготовки IT-кадров на базе кафедры системного программирования СПбГУ . . . . .	134
Наукоград и технопарки в Петергофе. . . . .	139
Национальные черты производства ПО. . . . .	142
Говорят коллеги: Игорь Агамирзян. . . . .	145
Говорят коллеги: Валентин Макаров . . . . .	147





## Часть I. Общие вопросы технологии программирования

### 1. Понятие технологии программирования

Технология — это набор правил, методик и инструментов, позволяющих наладить *производственный* процесс выпуска какого-либо продукта [1]. Разумеется, это определение нельзя назвать полным. Надо упомянуть процессы планирования, измерения, оценки качества, ответственность исполнителя и многое другое. Но в мою задачу не входит написание статьи для энциклопедии на слово «технология» — я дал какое-то представление о его значении и для начала разговора этого вполне хватит. Подчеркну только еще раз производственный характер понятия «технология». Почему-то именно слово «производство» сильно раздражает моих коллег-математиков, занимающихся программированием. «Ты бы еще технологию создания романа «Война и мир» создал». Недаром же известный специалист Дональд Кнут назвал свой знаменитый многотомный труд «Искусство программирования». Наконец, мне удалось выяснить отношение к этому понятию самого Д. Кнута. Несколько лет назад наш университет присвоил этому несомненно блистательному ученому звание почетного доктора. По такому случаю наш декан собрал для встречи с Д. Кнутом десять-двенадцать руководителей лабораторий и заведующих кафедрами, имеющих отношение к программированию. Дело было летом, Д. Кнут был с женой, переводчица поминутно смотрела на часы, напоминая, что гостям пора «на фонтаны», сам почетный доктор откровенно дремал, словом, это было скучнейшее мероприятие. Я решил пошутить, чтобы как-то развеять унылую атмосферу. И когда подошла моя очередь выступать, я вместо тридцатисекундного перечисления заслуг нашего коллектива сказал, что лично мне профессор Кнут доставил много неприятностей. Тот аж подскочил: «Я же в первый раз вас вижу!» Тут я объяснил, что много лет занимаюсь промышленной технологией программирования, а многие коллеги, включая и тех, кто сидит сейчас за этим столом, ссылаясь на его книгу, не признают это наукой, имеющей отношение к математике. Профессор очень разволновался, принял мою шутку вполне всерьез и пожаловался, что все это происходило и с ним: «Вы знаете, что я много лет занимался созданием и продвижением на рынок издательской системы TEX? Это что — не производство? Могу я на старости лет написать книжку с теоремами?»

В общем, я получил огромный заряд положительных эмоций.

Разумеется, я понимаю, что Д. Кнут известен скорее своими теоретическими работами, а ТЕХ — это скорее от безысходности, он задумал его после выпуска первого тома «Искусства программирования», поражаясь скудости рынка инструментальных средств для набора математических текстов, прервался на несколько лет для реализации ТЕХ и только потом продолжил свой титанический труд. Для него промышленное программирование — скорее бравада, что вот, мол, мы все можем, но все равно приятно получить поддержку от такого знаменитого ученого.

Мое первое столкновение с промышленностью состоялось в 1980-м году. К тому времени я уже несколько лет руководил лабораторией системного программирования, занимались мы самыми трудными проблемами, часто имеющими весьма отдаленное отношение к практике, но нас это несколько не беспокоило. «Мат-мех — лучше всех» — в глубине души я и сейчас в этом уверен. Так вот, в декабре 1980-го года меня пригласили в кабинет ректора ЛГУ, там было человек двадцать в военной форме и несколько партийных деятелей во главе с начальником оборонного отдела Обкома КПСС. Мне было сказано, что оборонная промышленность столкнулась с массой проблем при создании программного обеспечения (ПО) систем оборонного назначения, поэтому решено, что ЛГУ в лице нашей лаборатории должен помочь. Моего согласия никто не спрашивал, да в те времена этого и не требовалось. Так я начал работать с ЛНПО «Красная Заря», «Импульс», «Морфизприбор», «Ленинец», «Аврора», «Гранит» — ведущими предприятиями Ленинграда, работающими в интересах различных родов войск и ведомств. С большинством этих предприятий мы работаем до сих пор.

Нас использовали как «пожарную команду». Много лет сотрудники названных и других предприятий разрабатывали ПО для важных заказов, когда же все сроки (и деньги!) истекали, приглашали нас, чтобы мы навели порядок. Нельзя сказать, что до этого мы жили в башне из слоновой кости. У нас были договора с НИЦЭВТ по созданию транслятора с языка Алгол 68[2] для только что созданного (мы говорили «созданного») семейства ЕС ЭВМ, так что мы одними из первых освоили новую операционную систему OS/360. Пользуясь открытостью университета, к нам все время приходили какие-то люди (в том числе и военные) с просьбой найти ошибку в программе. Не было случая, чтобы мы искали ошибку больше двух дней.

Но все это не идет ни в какое сравнение с трудностями нашего понимания проблем реальной промышленности. Одно дело, когда у тебя в подчинении пятнадцать человек в университетской лаборатории, которых никогда не надо подгонять, сомневаться в их способностях, когда царит дух соревнования (не только между собой), когда все понимают друг друга с полуслова. Совсем другое дело — триста человек с совершенно

разными способностями и образованием, явно отсиживающие свои восемь часов, работающие в кодах (буквально с инженерного пульта ЭВМ), причем все это погружено в атмосферу секретности. Мне никогда не забыть, как одна женщина целый час объясняла мне схему распределения памяти, которую она вычитала у одного японского автора. С большим трудом удалось выяснить, что она имеет в виду стек. Другой программист запомнился мне тем, что для возврата из процедуры использовал регистр R7, хотя остальные триста человек для этой цели применяли R10. Он еще доказывал, что именно он прав, так как R7 – индексный регистр, поэтому передача управления по его содержимому выполняется на целую микросекунду быстрее. Мне так и не удалось объяснить ему важность соблюдения соглашения о связях.

Первыми из военных организаций к нам обратились сотрудники ЛНПО «Красная Заря» с просьбой помочь в программировании широкого класса задач управления и связи, в частности, в создании функционального программного обеспечения (ФПО) телефонных станций, управляемых специализированными ЭВМ (СЭВМ). Несколько лет ушло на изучение предметной области, пробные реализации и решение организационных вопросов. У сотрудников ЛГУ был исходный принцип важности использования алгоритмических языков высокого уровня (АЯВУ), основанный на опыте предыдущей работы. Однако начинать пришлось совсем не с этого, а с повышения общей культуры программирования разработчиков ФПО. Дело в том, что в области встроенного ФПО реального времени традиционно используются СЭВМ с нестандартной архитектурой, ориентированной на заданную предметную область (правда, неясно, в чем должна выражаться такая ориентация: например, если ЭВМ хорошо выполняет какие-то специальные операции, но плохо — условные переходы и вызовы процедур, которые встречаются гораздо чаще, можно ли считать, что данная ЭВМ соответствует предметной области?). Нестандартность архитектуры и малая тиражность таких СЭВМ приводят к отсутствию достаточно развитых операционных систем, трансляторов, средств отладки и других, ставших уже привычными, инструментов программирования. Поэтому мы столкнулись с работой на перфокартах и непосредственно за пультом СЭВМ «на тумблерах».

Мы пытались воспользоваться известными в то время технологиями, однако оказалось, что, например, P-технология [3] не имеет никаких средств настройки на СЭВМ, а предлагаемые в ней графический стиль программирования и программа-организатор с ручным вводом мало помогают в решении задач реального времени; технология РУЗА позволяет автоматизированным образом (но с большими доработками) построить кросс-ассемблер нужной ЭВМ и интерпретатор ее системы команд, а также осуществить некоторую регламентацию работы (например, стандартизовать имена

объектов). Р-технология была отвергнута практически сразу (за неимением конкретных программных средств на заданных нам СЭВМ), РУЗА в течение полугода была настроена на одну СЭВМ, однако полностью учесть все особенности СЭВМ так и не удалось; кроме того, параметрически настраиваемые кросс-ассемблер и интерпретатор замедляют работу в 5-7 раз.

Мы решили, что из-за таких простых программ, как ассемблер, не стоит «городить огород», и за короткое время реализовали новые кросс-ассемблер и интерпретатор, которые вместе с текстовым документатором и некоторыми сервисными программами составили основу первой внедренной нами в производство (1984 год) технологической системы, интенсивно использовавшейся сотнями разработчиков ФПО. Разумеется, мы отчетливо понимали ограниченность возможностей такой технологии, но ее популярность имела свои причины:

- применение вместо СЭВМ широко доступной ЕС ЭВМ с многопользовательской ОС и широкими сервисными возможностями;
- работа одновременно многих пользователей за терминалами, а не с перфокартами;
- богатые отладочные возможности интерпретатора СЭВМ, недостижимые непосредственно на СЭВМ;
- впервые осознанная разработчиками ФПО ценность документации на машинном носителе, легкость ее оформления, исправления и тиражирования;
- практически неограниченные возможности развития технологии, удивительно быстро схваченные разработчиками ФПО, в результате чего практически ежемесячно появлялись новые идеи и предложения, большинство из которых было быстро реализовано.

На протяжении всей совместной работы сотрудников ЛГУ и «Красной Зари» имело место противостояние двух позиций относительно понятия технологии программирования. Сотрудники ЛГУ, в основном, подразумевали под ним широкое использование инструментальных средств, а сотрудники «Красной Зари» настаивали на том, что технология — это, прежде всего, набор формальных методик и регламентирующих средств, позволяющих, в частности, на каждом этапе провести экспертизу, архивацию и измерение объема и качества проделанной работы. Такой подход вызывал постоянное раздражение профессиональных программистов.

Разумеется, мы понимали, что работа без четких сроков и ответственности перед соисполнителями возможна только исследовательская и экспериментальная, да и то в ограниченных объемах, но работа по конкретным заказам в промышленности дала массу новых впечатлений и импульсов для изучения. Например, уход исполнителя в самом разгаре работ. Оказалось, что в промышленности это довольно частое явление (отсюда необходимость архивации и других средств отчуждения результатов

работы от исполнителя); в коллективе из нескольких сот человек всегда есть люди, попросту не желающие работать, и хотя мы понимаем, что оценка работы программиста в байтах весьма сомнительна, это не отменяет необходимости учета индивидуально выполненной работы (измерение). Даже такое «приземленное» соображение, как разделение ответственности за принятые решения и ошибки, также нельзя сбрасывать со счетов (отсюда строгое документирование).

Особенно много разночтений вызывала необходимость оформления постановки задачи по стандартным правилам. Дело в том, что аккуратное оформление требует усилий, причем не только технического порядка, а программисту кажется, что, имея в руках такой мощный инструмент, как АЯВУ, легче сразу выразить свое понимание задачи в программе, а не в каких-то таблицах и диаграммах. Но непосредственное программирование лишает программиста обратной связи с функционалистом (постановщиком задачи) и уж тем более — с заказчиком. Большинство сложных систем невозможно сдать в эксплуатацию из-за огромного количества сравнительно мелких замечаний, вызванных разночтениями и неясностями в постановке задачи.

Мы настаивали на том, что, занимаясь вопросами документирования, ценообразования, способами регламентирования и контроля за ходом работ, нельзя забывать, что основным результатом применения технологии является программа, действующая в заданной вычислительной среде, хорошо отлаженная и документированная, доступная для понимания и развития в процессе сопровождения («нам нужны не приборы в принципе, а приборы в корпусе»).

Только после двух-трех лет работы в промышленности мы осознали, что нельзя все сводить к программному инструментарию. Поначалу мы с гневом отказывались от требований начальства детально документировать, кто, что и за какой период написал, но оказалось, что в большом коллективе всегда находятся милые в общении, всеми любимые организаторы всевозможных мероприятий, которые вообще ничего не делают по работе. Первую сдачу проекта для Управления правительственной связи КГБ я завалил, так как буквально перед самой сдачей кто-то стащил одну (!) перфокарту, а эти товарищи всегда начинали с чистой машины и полной перетрансляции. Вредителя так и не нашли, зато я получил хороший урок. Мы быстро реализовали контрольно-учетные программы, архивы с контролируемым доступом, многоуровневые системы сбора версий ПО и тому подобные «шпионские штучки». Так я впервые осознал разницу между «программированием для себя» и «программированием для хозяина», о которой так красочно рассказывал академик А. П. Ершов.

В более поздних публикациях эту разницу стали выражать более канцелярским стилем — просто программа и программный продукт.